Tampere University

Aki Kaipio

NO-REFERENCE IMAGE QUALITY ASSESSMENT WITH CONVOLUTIONAL NEURAL NETWORKS

Bachelor of Science Thesis
Faculty of Information Technology and Communication Sciences
April 2020

ABSTRACT

Aki Kaipio: No-reference image quality assessment with convolutional neural networks
Bachelor's thesis
Tampere University
Degree Programme in Computing and Electrical Engineering, BSc (Tech), Information Technology
April 2020

Software-based image quality assessment is important in situations where the photographer is unable to monitor quality, or the camera system is automated. By providing a numerical value on image quality, the system can adjust camera parameters according to the scene or alert the user about poor quality. This paper presents two different convolutional neural network-based methods for solving the problem of no-reference image quality assessment.

The first method is based on transfer learning. MobileNet architecture, originally designed for image classification, was modified to solve regression problems. High-resolution input images were converted to fit the input size of the network by using an algorithm that detects and crops an area of the most focused part in the full image. The method achieved a Spearman correlation coefficient value of 0.7804 between the estimated and true quality scores when KonIQ-10k dataset was used for training and testing.

The second method consists of two steps. First, a convolutional neural network analyzes the quality of small, randomly chosen patches of the full image, and forms an estimated score based on them. This score, and global features calculated from the full image are then used as input for a simple feedforward neural network, which outputs the final prediction. Examples of global features used in this work are brightness, noise and blur levels. With the same dataset, this method achieved a Spearman correlation coefficient value of 0.7270. The actual maximum accuracy of the method is higher than what was achieved in the work, as the number of features in this implementation was quite small.

Neither of the methods reached the accuracy of current top-performing methods, but the latter approach offers excellent customizability for different use cases. By adjusting the number and type of global features, the accuracy and performance of the method can be balanced to find an optimal configuration for each application.

Keywords: Image quality assessment, neural networks, transfer learning, MobileNet, KonIQ-10k

PREFACE

The topic for this bachelor's thesis work was suggested by Computational Imaging Group of Tampere University. Special thanks to Mykola Ponomarenko for supervising the work and supporting in difficult stages.

Tampere, 29 April 2020

Aki Kaipio

CONTENTS

.

1. INTRODUCTION

Since the invention of cameras, the technology has found its way to multiple applications. Nowadays, cameras are not only used for personal and professional photography, but also for surveillance, traffic monitoring and automated manufacturing processes. Achieving perfect image quality is a challenge in all applications. Digital cameras and smartphones have relatively small screens, so the user may not be able to see if the picture is noisy, slightly out of focus or contains some other distortions. Automated cameras must often adapt to constantly changing lighting conditions and to focus on relevant points in the scene.

By providing a reliable metric on image quality, cameras can self-adjust their parameters to be able to capture the best possible picture, or the software could alert the user to retake a picture if the previous one had poor quality. Full reference-based metrics, like structural similarity (SSIM) and feature similarity (FSIM) correlate well with subjective quality when a reference image with ideal quality is available for comparison [1]. This is often not the case in real-world applications, where the scene is constantly changing. Therefore, a no-reference method must be used instead.

In this work, two different convolutional neural network (CNN) based methods are used to approach the problem of no-reference image quality assessment (NR-IQA). The first method uses transfer learning by modifying a pre-trained MobileNet model originally designed for image classification. The second method uses a simple custom network to analyze small random patches of an image and calculates estimated quality for the full image from them. The accuracy is then further improved by using that score in a second neural network together with global characteristics, like brightness, saturation and estimated noise level.

Chapter 2 introduces earlier work on NR-IQA with CNNs. Chapter 3 explains the details of the dataset used in this work. Chapters 4 and 5 explain the two methods in detail. Chapter 6 compares the two methods.

2. RELATED WORK

Multiple different methods based on convolutional neural networks have been applied to the task of NR-IQA. Wiedemann et al. proposed a method that generates a quality map from an image by analyzing it in small patches. CNN takes inputs of a size of 64x64 pixels, and the full image is scanned in sliding window fashion. Further improvements were made by augmenting sharpness and brightness information into quality maps. [2]

Multiple transfer learning -based methods have also been researched. Otroshi-Shahreza et al. have used pre-trained Xception network as a basis and added custom layers to output the final score distribution. [3]

Hosu et al. experimented with multiple base network architectures available in Python Keras library (VGG16, ResNet101, InceptionV3, InceptionResnetV2 and NASNetMobile). They also tested each network with different input sizes and loss functions. The best performing model, which used InceptionResnetV2 with pre-trained weights as the base network, achieved a Spearman's rank order correlation coefficient (SROCC) of 0.921 between predicted scores and mean opinion scores of KonIQ-10k dataset. [4]

3. KONIQ-10K DATASET

KonIQ-10k is a subset of Yahoo Flickr Creative Commons 100 Million (YFCC100m) dataset, targeted for researchers developing image quality assessment methods. Authors have algorithmically chosen 10073 images from YFCC100m such, that they have large variety of content and distortions. Original images have been rescaled and cropped to a resolution of 1024x768 pixels, but to ensure authenticity, no artificial distortion generation has been applied. [4]

Each image has been subjectively rated in scale of 1 to 5 by carefully selected crowd workers. From those scores, a mean opinion score (MOS) has been calculated for each image. [4]

MOS = 1.0962 MOS = 3.4231 MOS = 4.3100

Figure 1. *Sample images from KonIQ-10k dataset with their mean opinion scores.*

For this thesis work, KonIQ-10k dataset was divided into training and testing sets, with 8000 images belonging to the training set, and the remaining 2073 images to the testing set. MOS was used as a ground truth for image quality.

4. TRANSFER LEARNING WITH MOBILENET ARCHITECTURE

MobileNets are neural networks that utilize depthwise separable convolutions to reduce computational cost and memory footprint compared to standard convolutions. Input size and number of channels in MobileNet network can also be scaled down, so the network can be adjusted to achieve ideal balance between execution speed, size and prediction accuracy. [5]

Table 1. *MobileNet Body Architecture [5].*

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

The original MobileNet architecture has been designed for image classification, so to work for IQA, some modifications were applied. After the average pooling layer, two fully connected layers were used, with 1024 neurons each. The final layer is a fully connected layer with a single neuron to output the final quality score.

4.1 Input image pre-processing

Original images in KonIQ-10k dataset have a resolution of 1024x768 pixels, which must be reduced to match the input size of MobileNet network. Maximum input size of the network, which is 224x224 pixels, was selected to be used in this work to preserve as much information from the full image as possible.

There are multiple ways to reduce resolution, downscaling being one of the most common method, especially in image classification tasks. But for image quality assessment it is not a good option, as it removes noise. Cropping a patch is preferable, as it preserves fine details that are important in quality analysis.

Instead of just cropping the central region of each image, a more sophisticated method was used in this work. The method is based on detecting the most focused region (MFR) from full image by using the following algorithm:

1. Input image is converted to grayscale, and multiple downscaled versions are generated from it. Each resulting image is assigned a weight, which decreases with scaling factor.
2. Sharpness is measured by sum of local variances. The calculation is performed for each region with a sliding window of 3x3 pixels in size. Central regions are given bigger weights.
3. Resulting matrix is filtered with an average filter of window size (224x224 pixels in this case).
4. A patch that corresponds to the maximum value in filtering results is cropped from the original RGB image.

When the MFR patch is cropped from the image, most of the spatial information is lost. This may lead to unrealistic quality estimations because most of the unwanted blur, noise and other distortions are often in the background, outside the most focused region. Because of that, another set of input images were generated, where patches from outside the MFR are also included. In those composite images, the size of MFR patch is only 128x128 pixels and the remaining part is covered with patches picked from edges.

Figure 2. *Extraction of MFR patch (top) and generation of composite image (bottom)*

4.2 Training and testing

The MobileNet model provided by TensorFlow Keras API has an option to initialize weights randomly or from pre-trained weights. Pre-trained weights have been obtained from training on ImageNet dataset. [6] In this work, both options were tested to see how they compare.

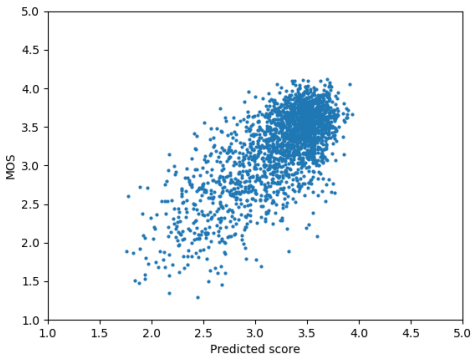
All network configurations were trained with same parameters. Mini-batch size was set to 32, 20% of training images were used for validation and each network configuration was trained for 100 epochs. Mean squared error (MSE) was used as loss function and the checkpoint with the smallest validation loss was saved as the final model. Adam optimizer was used with default parameters (learning_rate=0.001, beta_1=0.9, beta_2=0.999, epsilon=1e-07, amsgrad=False).

The following table shows Spearman's rank order correlation coefficients between estimated quality and mean opinion scores for each network configuration.

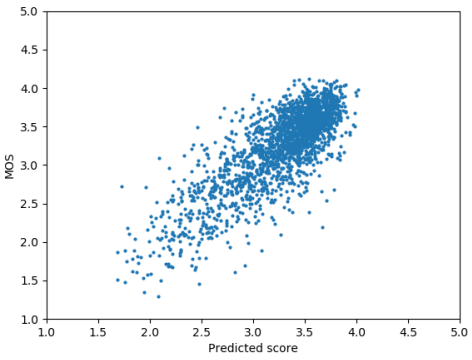
Table 2. *Test results of different MobileNet configurations.*

Initialization of weights	**Input images**	**SROCC**
Random	MFR	0.6047
Random	MFR with edge patches	0.6457
Pre-trained	MFR	0.7366
Pre-trained	MFR with edge patches	0.7804

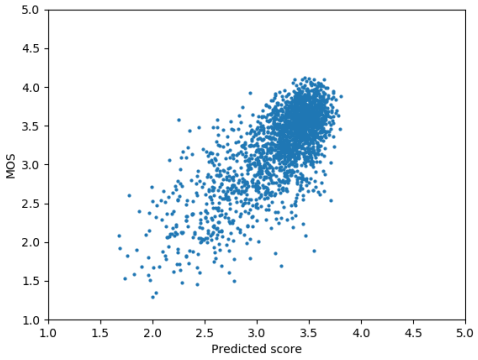
Test results show a big difference in SROCC between models that used pre-trained weights and the ones that did not. Input image type had smaller, but still significant effect in SROCC.



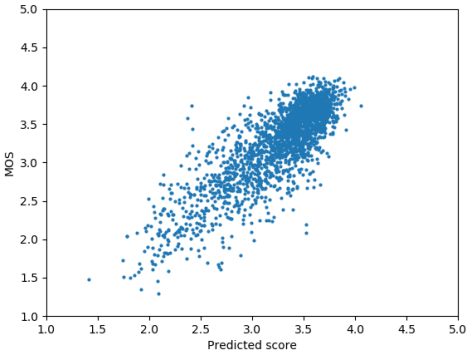
Random, MFR



Pre-trained, MFR



Random, MFR with edge patches



Pre-trained, MFR with edge patches

Figure 3. *Scatter plots of test results with different configurations (weight initialization, input image type)*

5. QUALITY PREDICTION FROM MINI-PATCHES AND GLOBAL FEATURES

To avoid the shortcomings of the first method, like reliance from pre-trained weights, a second method was developed. It is based on two individual steps. The first step consists of a convolutional neural network which analyzes quality of small patches randomly selected from an image. In the second step, quality estimations acquired from the first step are used together with global features in a feedforward neural network to output the final score.

5.1 Mini-patch quality estimation with CNN

Two different sizes for input patches were used, 16x16 pixels and 32x32 pixels. Convolution size was 3x3 pixels for the former and 5x5 pixels for the latter input size. Otherwise, the network structures were identical, consisting of the input layer and 9 convolutional layers separated by ReLU layers. Each convolutional layer contained 128 feature maps, except the last one, which outputs a single value as quality prediction.

Also, two different loss functions were used to test if one is more suitable for the task than the other. As predicted score has linear correlation with MOS in the ideal case, Pearson correlation can be used as loss function. Because perfect correlation should result in zero loss, the Pearson correlation is subtracted from 1 according to the following equation:

$$Loss = 1 - \rho_{X,Y} = 1 - \frac{cov(X,Y)}{\sigma_X \sigma_y}$$

where $\rho_{X,Y}$ is Pearson's correlation coefficient, X is predicted score, Y is MOS, σ_X is standard deviation of X and σ_Y is standard deviation of Y.

Each network was trained for 3600 iterations with a batch size of 500. Initial learning rate was 0.0001 and it was updated on each epoch according to the equation

$$r = \frac{r_{initial}}{1 + di}$$

where r is learning rate, d is decay (0.0001 is this case) and i is the number of current iteration.

To estimate the quality of a full image, 100 random patches are cropped from it. Each patch is then evaluated with the network and results are stored in a vector. By analyzing

the vectors, it was found that the highest correlation between patch predictions and MOS occurred with 95th percentile predictions. Therefore, that is used as the final prediction for the full image.

Table 3. *Test results from trained network.*

Input size	**Loss function**	**SROCC**
16x16	MSE	0.5804
16x16	1 - Pearson correlation	0.6032
32x32	MSE	0.6043
32x32	1 - Pearson correlation	0.6484

Results show that mini-patch size of 32x32 pixels provided more accurate predictions than 16x16 pixels, and that Pearson correlation as loss function improves accuracy compared to MSE.

5.2 Improving the accuracy with global features

The first network only analyzes image quality of small patches. It cannot see the full image and spatial content, which significantly restricts the accuracy. To fix these limitations, some global features of the image were calculated to be used as input in a second neural network together with scores from the first step.

There are countless features and methods for calculating them, and it is impossible to include everything in a single network. In this work, 19 different features were chosen to test the potential of this method.

Features were calculated for the original full-sized image and for the MFR patch. This improves the accuracy of the network, as some features, like the amount of blur has significantly different values and correlations with MOS in MFR patch and full image.

The following table describes all features and Spearman’s rank order correlation coefficients with mean opinion scores.

***Table 4**. Global features used in this implementation*

	SROCC with MOS when the feature is calculated from	
Feature	**Full image**	**MFR patch**
blur	-0.2474	-0.4952
mean saturation	-0.2519	-0.2517
median saturation	-0.2560	-0.2602
mean luminance	0.2783	0.2576
median luminance	0.3418	0.2337
luminance range	0.3237	0.2101
mean chrominance (Cb)	0.0827	-0.0103
median chrominance (Cb)	-0.0548	-0.0216
chrominance range (Cb)	-0.0517	-0.0302
mean chrominance (Cr)	0.0196	-0.0572
median chrominance (Cr)	-0.0662	-0.0591
chrominance range (Cr)	-0.0780	-0.0572
noise variance (red channel)	0.4618	0.5773
noise variance (green channel)	0.4625	0.5770
noise variance (blue channel)	0.4613	0.5779
mean absolute value of FFT	0.5933	0.6143
median absolute value of FFT	0.5925	0.6296
minimum absolute value of FFT	0.4269	0.4652
maximum absolute value of FFT	0.2746	0.0850

Blur level estimations were calculated with the code written by Do Quoc Bao [7], and the method itself is described in the paper by Crété-Roffet et al. [8].

Noise variance was calculated by using a Laplacian mask-based method suggested by John Immerkaer [9]. The actual code was written by Tolga Birdal [10].

As can be seen from the table, some features, like median absolute value of FFT for MFR patch provided correlations close to the scores from the first network. On the other end, chrominance-based values correlated very little with MOS.

5.3 Final prediction with feedforward network

In the final step, 95th percentile score from the convolutional neural network and all 19 global features for full images and MFR patches were used to train the network for final quality prediction. A simple feedforward neural network was used for this task. After experimenting with different numbers of layers and number of neurons per layer, the optimal network size was found to be 1 hidden layer with about 16 neurons. Small difference in number of neurons did not have much effect on accuracy.

Three different input configurations were used to see how the addition of second network affects the final SROCC. In the first configuration, only global features were used as input. The other two configurations include quality predictions from the two best-performing models from the first step.

As the feedforward network is so small, there are slight differences in SROCC each time the network is retrained. To get more reliable value for the accuracy, each network was trained and tested 10 times and the mean SROCC was saved.

Table 5. *Final SROCCs from the two-step method.*

CNN	**Mean SROCC**
None (use only global features)	0.7167
32x32 MSE	0.7185
32x32 Pearson	0.7270

Final test results show that the global features themselves can provide significantly better accuracy than the mini-patch-based method. However, output from the first network with the best configuration achieved better correlation than any of the global features, and by including it in the input, the accuracy increased to 0.7270.

It should be noted that the correlation achieved in this work is not the maximum this method is capable of. By including more global features with different algorithms, the accuracy will get better. The downside is the increase in execution speed, which at some point will make the method unpractical for real-world usage.

6. COMPARISON OF METHODS

The best result, with a SROCC of 0.7804, was achieved with MobileNet based model. In addition to better accuracy, it is also easier to implement and has better overall performance compared to the second method.

Its most significant limitation is the dependency from pre-trained weights, as SROCC dropped to 0.6457 if weights were initialized randomly. Another drawback is higher memory consumption and loading time. The network variant used in this work has 5,329,089 parameters, compared to 2,730,497 in the CNN of the second method.

Biggest strength of the second method is customizability. Global features can be chosen separately for each application and use case, and their number can be adjusted to achieve optimal balance between accuracy and performance.

Usage of the CNN for analyzing small patches is optional, as its contribution to final accuracy is relatively small. It was the most resource-heavy component of this method, so leaving it out will increase the execution speed, which in current configuration is much slower than with the MobileNet-based method.

7. CONCLUSIONS

This thesis work proposed two different methods that can be used for NR-IQA. The first method uses transfer learning to make Keras MobileNet model, originally designed and trained for image classification, to perform image quality assessment. The second method uses a custom convolutional neural network to analyze quality of mini-patches, and together with global features, estimates the quality of full-sized images.

The accuracies of proposed methods are lower compared to top-performing CNN-based NR-IQA methods trained and tested with the same dataset. However, the second proposed method provides excellent customizability and it can be fine-tuned to meet different requirements on accuracy and performance.

Future research could be focused into global feature extraction in the second method. In this work, only a small number of features were tested, and the full potential is still undiscovered.

REFERENCES

[1] Samajdar T, Quraishi I. Analysis and evaluation of image quality metrics. In: Advances in Intelligent Systems and Computing. Springer Verlag; 2015. p. 369–78.

[2] Wiedemann O, Hosu V, Lin H, Saupe D. Disregarding the Big Picture: Towards Local Image Quality Assessment. In: 2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX). IEEE; 2018. p. 1–6.

[3] Otroshi-Shahreza H, Amini A, Behroozi H. No-Reference Image Quality Assessment using Transfer Learning. In: 2018 9th International Symposium on Telecommunications (IST). IEEE; 2018. p. 637–40.

[4] Hosu V, Lin H, Sziranyi T, Saupe D. KonIQ-10k: An Ecologically Valid Database for Deep Learning of Blind Image Quality Assessment. IEEE Transactions on Image Processing. 2020; 29:4041–56.

[5] Howard A, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv.org [Internet]. 2017 Apr 17; Available from: http://search.proquest.com/docview/2074236657/

[6] Keras Documentation, Applications, Available from: https://keras.io/applications/, Accessed 27 April 2020

[7] Do Quoc Bao (2020). Image Blur Metric (https://www.mathworks.com/matlabcentral/fileexchange/24676-image-blur-metric), MATLAB Central File Exchange. Retrieved April 26, 2020

[8] Crété-Roffet, F, Dolmière, T, Ladret, P, Nicolas, Marina. The Blur Effect: Perception and Estimation with a New No-Reference Perceptual Blur Metric. 2007. Human Vision and Electronic Imaging. 12. 10.1117/12.702790.

[9] Immerkaer J. Fast noise variance estimation. CVIU: Computer Vision and Image Understanding [Internet]. 1996 Jan 1;64(2):300–2. Available from: http://search.proquest.com/docview/23742737/

[10] Tolga Birdal (2020). Fast Noise Estimation in Images (https://www.mathworks.com/matlabcentral/fileexchange/36941-fast-noise-estimation-in-images), MATLAB Central File Exchange. Retrieved April 27, 2020.